Amendments to the Specification:

Please replace title with the following amended title:

APPARATUS FOR ~~GENERATING AND IMPLEMENTING A COMMUNICATION PROTOCOL AND INTERFACE FOR~~ HIGH DATA RATE SIGNAL TRANSFER USING A COMMUNICATION PROTOCOL

Please replace paragraph [0001] with the following amended paragraph:

The present Application for Patent claims priority to Provisional Application No. 60/255,833 filed December 15, 2000 and Provisional Application No. [[60/]]60/317,858 filed September 6, 2001, and assigned to the assignee hereof and hereby expressly incorporated by reference herein.

Please replace paragraph [0002] with the following amended paragraph:

The present Application for Patent is a continuation and claims priority to Patent Application No. 10/020,520 entitled "GENERATING AND IMPLEMENTING A COMMUNICATION PROTOCOL AND INTERFACE FOR HIGH DATA RATE SIGNAL TRANSFER" filed December 14, 2001, now Pat. No. 6,760,772 ~~pending~~, and assigned to the assignee hereof and hereby expressly incorporated by reference herein.

Please replace paragraph [0018] with the following paragraph amended to remove highlighting:

Embodiments for the invention are directed to a Mobile Digital Data Interface (MDDI) for transferring digital data at a high rate between a host device and a client device over a communication path which employees a plurality or series of packet structures linked together to form a communication protocol for communicating a pre-selected set of digital control and presentation data between the host and client devices. The signal communications protocol or link layer is used by a physical layer of host or client link controllers. At least one link controller residing in the host device is coupled to the client device through the communications path or link, and is configured to generate, transmit, and receive packets forming the communications protocol, and to form digital presentation data into one or more types of data packets. The interface

provides for bi-directional transfer of information between the host and client.

Please replace paragraph [0031] with the following amended paragraph:

FIG. 5 illustrates the use of an MDDI link controller and the types of signals passed between a host and a client over the physical data link conductors for Types II, [[II]] III, and IV interfaces.

Please replace paragraph [0145] with the following amended paragraph:

The Sub-frame Length field contains 4 bytes of information that specifies the number of bytes per sub-frame. The length of this field may be set equal to zero to indicate that only one sub-frame will be transmitted by the host before the link is shut down into an idle state. The value in this ~~filed~~ field can be dynamically changed "on-the-fly" when transitioning from one sub-frame to the next. This capability is useful in order to make minor timing adjustments in the sync pulses for accommodating isochronous data streams. If the CRC of the Sub-frame Header packet is not valid then the link controller should use the Sub-frame Length of the previous known-good Sub-frame Header packet to estimate the length of the current sub-frame.

Please replace paragraph [0150] with the following amended paragraph:

Video Stream Packets carry video data to update atypically rectangular region of a display device. The size of this region may be as small as a single pixel or as large as the entire display. There may be an almost unlimited number of streams displayed simultaneously, limited by system resources, because all context required to display a stream is contained within the Video Stream Packet. The format of the video stream packet (Video Data Format Descriptor) is shown in FIG. 10. As seen in FIG. 10, this type of packet is structured to have Packet Length, Packet Type, Video Data Format Descriptor, Display Attributes, X Left Edge, Y Top Edge, X Right Edge, Y Bottom Edge, X and Y Start, Pixel Count, Parameter CRC, Pixel Data, and CRC fields. This type of packet is generally identified as a Type 1, which is indicated in the 1 byte type field.

Please replace paragraph [0153] with the following amended paragraph:

FIGS. 11a through 11d illustrate how the Video Data Format Descriptor is coded. As used in these figures, when bits [15:13] are equal to '000', as shown in FIG. 11a, then the video data consists of an array of monochrome pixels where the number of bits per pixel is defined by bits 3 through 0 of the Video Data Format Descriptor word. Bits 11 through 4 are set to zero in this situation. When bits [15:13] are instead equal to '001', as shown in FIG. 11b, then the video data consists of an array of color pixels that each specify a color through a color map. In this situation, bits 5 through 0 of the Video Data Format Descriptor word define the number of bits per pixel, and bits 11 through 6 are [[be]] set equal to zero. When bits [15:13] are instead equal to '010', as shown in FIG. 11c, then the video data consists of an array of color pixels where the number of bits per pixel of red is defined by bits 11 through 8, the number of bits per pixel of green is defined by bits 7 through 4, and the number of bits per pixel of blue is defined by bits 3 through 0. In this situation, the total number of bits in each pixel is the sum of the number of bits used for red, green, and blue.

Please replace paragraph [0166] with the following amended paragraph:

A host needs to know the capability of the display (client) it is communicating with in order to configure the host-to-display link in [[an]] a generally optimum or desired manner. It is recommended that a display send a Display Capability Packet to the host after forward link synchronization is acquired. The transmission of such a packet is considered required when requested by the host using the Reverse Link Flags in the Reverse Link Encapsulation Packet. The format of the Display Capability packet is illustrated in FIG. 18. As shown in FIG. 18, this type of packet is structured to have Packet Length, Packet Type, Protocol Version, Min Protocol Version, Bitmap Width, Bitmap Height, Monochrome Capability, Color Map Capability, RGB Capability, Y Cr Cb Capability, Display Feature Capability, Data Rate Capability, Frame Rate Capability, Audio Buffer Depth, Audio Stream Capability, Audio Rate Capability, Min Sub-frame rate, and CRC fields. This type of packet is generally identified as a Type 66 packet.

Please replace paragraph [0167] with the following amended paragraph:

A keyboard data packet is used to send keyboard data from the client device to the host. A wireless (or wired) keyboard may be used in conjunction with various displays or audio ~~deices~~ devices, including, but not limited to, a head mounted video display/audio presentation device. The Keyboard Data Packet relays keyboard data received from one of several known keyboard-like devices to the host. This packet can also be used on the forward link to send data to the keyboard. The format of a Keyboard Data Packet is shown in FIG. 19, and contains a variable number of bytes of information from or for a keyboard. As shown in FIG. 19, this type of packet is structured to have Packet Length, Packet Type, Keyboard Data, and CRC fields. This type of packet is generally identified as a Type 67 packet.

Please replace paragraph [0182] with the following amended paragraph:

The Perform Type Handoff Packet is a means for the host to command the display to handoff to the mode specified in this packet. This is to be the same mode that was previously requested and acknowledged by the Interface Type Handoff Request Packet and Interface Type Acknowledge Packet. The host and display should switch to the agreed upon mode after this packet is sent. The display may lose and re-gain link synchronization during the mode change. The format of a Perform Type Handoff Packet is shown in FIG. 29. As shown in FIG. 29, this type of packet is structured to have Packet Length, Packet Type, ~~Packet~~ Interface Type, and CRC fields. This type of packet is generally identified as a Type 77 packet in the 1-byte type field, and uses a pre-selected fixed length of 4 bytes.

Please replace paragraph [0190] with the following amended paragraph:

In order to accurately determine the round trip delay time for signals traversing to and from the client, the host counts the number of bit time periods occurring after the start of the Measurement Period until the beginning of the 0xff, 0xff, 0x0 sequence is detected upon arrival. This information is used to determine the amount of time for a round trip signal to pass from the host to the client and back again. Then, about one half of this amount is attributed to [[a]] the delay created for the one way passage of a signal to the client.

Please replace paragraph [0241] with the following amended paragraph:

Depending on the forward link data rate and signal processing delays encountered, it may require more time than one cycle on the MDDI_Stb signal for this "round trip" effect or set of events to be completed, which results in the consumption of undesirable amounts of time or cycles. To circumvent this problem, the Reverse Rate Divisor makes it possible for one bit time on the reverse link to span multiple cycles of the MDDI_Stb signal. This means that the reverse link data rate is less than the forward link rate.

Please replace paragraph [0251] with the following amended paragraph:

The reverse link clock for the host is at zero until the end of the Turn Around 1 period, when the clock is started to accommodate the reverse link packets The arrows in the lower portion of the figure indicate when the data is sampled, as would be apparent from the remainder of the disclosure. The first byte of the packet field being transferred (here 11000000) is shown commencing after Turn Around 1 and the line level has stabilized from the host driver being disabled. The delay in the passage of the first bit, and as seen for bit three, can ~~bee~~ be seen in the dotted lines for the Data signal.

Please replace paragraph [0263] with the following amended paragraph:

A summary of the general steps undertaken in processing data and packets during operation of an interface using embodiments of the invention is shown in FIGS. 54a and 54b, along with an overview of the interface apparatus processing the packets in FIG. 55. In these figures, the process starts in a step 5402 with a determination as to whether or not the client and host are connected using a communication path, here a cable. This can occur through the use of periodic polling by the host, software or hardware that detects the presence of connectors or cables or signals at the inputs to the host (such as is seen for USB interfaces), or other known techniques. If there is no client connected to the host, then it can simply enter a wait state of some predetermined length, depending upon the application, go into a hibernation mode, or be inactivated to await future use which might require a user to take action to reactive the host. For example, when a host resides on a computer type device, a user might have to click on an on screen icon or request a program that activates the host processing ~~tot~~ to look for the client. Again, simple plug in of a USB type

connection, such as used of the Type-U interface, could activate host processing.

Please replace paragraph [0264] with the following amended paragraph:

Once a client is connected to the host, or ~~visa~~ vice versa, [[,]] or detected as present, either the client or the host sends appropriate packets requesting service in steps 5404 and 5406. The client could send either Display Service Request or Status packets in step 5404. It is noted that the link, as discussed above, could have been previously shut down or be in hibernation mode so this may not be a complete initialization of the communication link that follows. Once the communication link is synchronized and the host is trying to communicate with the client, the client also needs to provide Display Capabilit~~ies~~y packets to the host, as in step 5408. The host can now begin to determine the type of support, including transfer rates, the client can accommodate.

Please replace paragraph [0267] with the following amended paragraph:

During operation one of several different events can occur which lead to the host or client desiring a different data rate or type of interface mode. For example, a computer or other device communicating data could encounter loading conditions in processing data that cause[[s]] a slow down in the preparation or presentation of packets. A display receiving the data could change from a dedicated AC power source to a more limited battery power source, and either not be able to transfer in data as quickly, process commands as readily, or not be able to use the same degree of resolution or color depth under the more limited power settings. Alternatively, a restrictive condition could be abated or disappear allowing either device to transfer data at higher rates. This being more desirable, a request can be made to change to a higher transfer rate mode.

Please replace paragraph [0271] with the following amended paragraph:

The packets being transferred in the above operations processing will be transferred using the drivers and receivers previously discussed in relation to the host and client controllers. These line drivers and other logic elements are connected to the state machine and general processors discussed

above, as illustrated in the overview of FIG. 55. In Fig. 55, a state machine 5502 and general processors 5504 and 5508 may further be connected to other elements not shown such as a dedicated USB interface, memory elements, or other components residing outside of the link controller with which they interact, including, but not limited to, the data source, and video control chips for viewing display devices.

Please replace paragraph [0272] with the following amended paragraph:

The processors, and state machine provide control over the enabling and disabling of the drivers as discussed above in relation to guard times, and so forth, to assure efficient establishment or termination of the communication link, and transfer of packets.

Please replace paragraph [0286] with the following amended paragraph:

The *Color Map Data Size field* (2 bytes) specifies the total number of color map table entries that exist in the Color Map Data in this packet. The number of bytes in the Color Map Data is 3 times the Color Map Size. The Color Map Size is [[et]] set to zero to send no color map data. If the Color Map Size is zero then a Color Map Offset value is still sent but it is ignored by the display. The *Color Map Offset field* (2 bytes) specifies the offset of the Color Map Data in this packet from the beginning of the color map table in the display device.

Please replace paragraph [0324] with the following amended paragraph:

The *Interface Type field* (1 byte) specifies the new interface type to use. The value in this field specifies the interface type in the following manner. If the value in Bit 7 is equal to 0 the Type handoff request is for the forward link, if it is equal to 1, then the Type handoff request is for the reverse link. Bits 6 through 3 are reserved for future use, and are generally set to zero. Bits 2 through 0 are used to define the interface Type to be used, with a value of 1 meaning a handoff to Type-I mode, value of 2 a handoff to Type-II mode, a value of 3 a handoff to Type-III mode, and a value of 4 a handoff to Type-IV mode. The values of 0 and 5 through 7 are reserved for future designation of alternative modes or combinations of modes.[[.]]

Please replace paragraph [0325] with the following amended paragraph:

The *Interface Type field* (1 byte) has a value[[s]] that confirms the new interface type to use. The value in this field specifies the interface type in the following manner. If Bit 7 is equal to 0 the Type handoff request is for the forward link, alternatively, if it is equal to 1 the Type handoff request is for the reverse link. Bit positions 6 through 3 are currently reserved for use in designating other handoff types, as desired, and are generally set to zero. However, bit positions 2 through 0 are used to define the interface Type to be used with a value of 0 indicating a negative acknowledge, or that the requested handoff cannot be performed, values of 1, 2, 3, and 4 indicating handoff to Type-I, Type-II, Type-III, and Type-IV modes, respectively. Values of 5 through 7 are reserved for use with alternative designations of modes, as desired.

Please replace paragraph [0327] with the following amended paragraph:

The *Audio Channel Enable Mask field* (1 byte) contains a group of flags that indicate which audio channels are to be enabled in a client. A bit set to one enables the corresponding channel, and a bit set to zero disables the corresponding channel Bits 0 through 5 designate channels 0 through 5 which address left front, right front, left rear, right rear, front center, and sub-woofer channels, respectively. Bits 6 and 7 are reserved for future use, and in the mean time [[are]] may be set to zero.